CTL, the branching-time temporal logic

Cătălin Dima

Université Paris-Est Créteil
Temporal properties

- Safety, termination, mutual exclusion – LTL.
- Liveness, reactivity, responsiveness, infinitely repeated behaviors – LTL.
- Available choices, strategies, adversarial situations?

CNIL

*Tout utilisateur peut demander le retrait de ses données...*

- How do we interpret *peut*?
  - $p = \text{demander le retrait...}$
  - Then formula $= \Box p$?
  - NO!

Strategy to win a game

Black has a strategy to put the game in a situation from which White king will never get close to Black pawn.

- Not specifiable in LTL either!
Computational Tree Logic (CTL)

Syntax:

\[ \Phi ::= p \mid \Phi \land \Phi \mid \neg \Phi \mid \forall \, \bigcirc \, \Phi \mid \forall \, \Box \, \Phi \mid \forall (\Phi \, U \, \Phi) \mid \exists \, \bigcirc \, \Phi \mid \exists \, \Box \, \Phi \mid \exists (\Phi \, U \, \Phi) \]

- **Grammar** for the logic: the set of formulas is the set of “words” obtained by this (context-free!) grammar, with \( \Phi \) viewed as nonterminal.

- **Syntactic tree** for each formula.
  - \( \forall, \exists \): path quantifier (will see why!).
  - \( U, \Box, \Diamond \): temporal quantifiers.
  - Alternative notations (for the temporal operators): \( \Box \phi = G\phi \), \( \Diamond \phi = F\phi \), \( \bigcirc \phi = X\phi \).
  - Each path quantifier must be followed by a temporal quantifier in the syntactic tree of each formula.

- **Sample formula**: \( p \land \exists \Box (\neg \forall \bigcirc p \lor \forall (p \, U \, (\neg q \land \exists \bigcirc q))) \).
  - Draw its syntactic tree!

- **Strict** alternation:
  - A non-CTL formula \( p \land \exists \Box (\neg \forall \bigcirc p \lor (p \, U \, (\neg q \land \exists \bigcirc q))) \).
  - ... because the \( U \) is not preceded by a path quantifier.
**Intuitive meanings:**

- $\forall \bigcirc p$: in any next state $p$ holds.
  
  *Regardless of the actions of the “environment”, at the next clock tick $p$ holds.*

- $\forall \square p$: $p$ will perpetually hold in any continuation from the current state.
  
  *Whatever the environment does, $p$ will hold forever.*

- $\forall p U q$: in any continuation from the current state $q$ eventually holds, and until then $p$ must hold.
CTL formulas

- Derived operators:

\[
\exists \lozenge \phi = \neg \forall \square \phi
\]
\[
\forall \lozenge \phi = \forall (true \cup \phi)
\]
\[
\exists \square \phi = \neg \forall \lozenge \neg \phi
\]
\[
\exists \bigcirc \phi = \neg \forall \square \phi
\]
\[
\exists (\phi \cup \psi) = \neg \forall (\neg \phi \cup (\neg \phi \land \neg \psi)) \land \neg \forall \square \psi
\]

- Some intuitive meanings:

  - \(\exists \lozenge \phi\): there exists a next state in which \(p\) holds.
    
    *The environment could make it possible for \(p\) to hold at the next clock tick.*
  
  - \(\exists \square \phi\): there exists a continuation on which \(p\) holds perpetually.
  
  - \(\forall \lozenge \phi\): in all continuations \(p\) eventually holds.
    
    *There is a guarantee that \(p\) must eventually hold, whatever the environment does.*
Branching time

The root in the following tree satisfies $\forall \circ p$:

The root in the following tree satisfies $\exists \circ p$:
Branching time, contd.
Transition systems

\[ \mathcal{T} = (Q, \Pi, \delta, \pi, q_0) \]

- \( Q \) finite set of states.
- \( \Pi \) finite set of atomic propositions.
- \( q_0 \in Q \) initial state.
- \( \delta \subseteq Q \times Q \) transition relation.
- \( \pi : Q \to 2^\Pi \) state labeling.

Example: the hunter/wolf/goat/cabbage puzzle.

- Nondeterminism: given \( q \in Q \), there may exist several \( r_1, r_2, \ldots \in Q \) with \( (q, r_1) \in \delta, (q, r_2) \in \delta \ldots \).

- Who chooses which successor in each state?
  - CTL answer: the environment does!
CTL semantics in transition systems

Recursively interpret each CTL formula in each state of the system

Given $T = (Q, \Pi, \delta, \pi, q_0)$ and $q \in Q$:

- $q \models p$ if $p \in \pi(q)$.
- $q \models \phi_1 \land \phi_2$ if....
- $q \models \neg \phi$ if...
- $q \models \forall \pi \phi$ if for all $r \in Q$ with $(q, r) \in \delta$, $r \models \phi$. Example:
CTL semantics in transition systems
Recursively interpret each CTL formula in each state of the system

Given $\mathcal{T} = (Q, \Pi, \delta, \pi, q_0)$ and $q \in Q$:

- $q \models p$ if $p \in \pi(q)$.
- $q \models \phi_1 \land \phi_2$ if....
- $q \models \neg \phi$ if...
- $q \models \forall \phi$ if for all $r \in Q$ with $(q, r) \in \delta$, $r \models \phi$. Example:
CTL semantics in transition systems (contd.)

Given $T = (Q, \Pi, \delta, \pi, q_0)$ and $q \in Q$:

- $q \models \forall \Box \phi$ if for each run $\rho$ in $T$ starting in $q$ with
  $\rho = q = q_0 \rightarrow q_1 \rightarrow \ldots \rightarrow q_n \rightarrow \ldots$ (infinite!) we have that $q_n \models \phi$ for all $n$.
  - In other words, $\rho \models \Box \phi$!

- $q \models \forall (\phi_1 U \phi_2)$ if for each run $\rho$ in $T$ starting in $q$ with
  $\rho = q = q_0 \rightarrow q_1 \rightarrow \ldots \rightarrow q_n \rightarrow \ldots$ there exists $n \geq 0$ with $q_n \models \phi_2$ and for all $0 \leq m < n$, $q_m \models \phi_1$.
  - In other words, $\rho \models \phi_1 U \phi_2$!
Property specification

**CNIL**

*Tout utilisateur peut demander le retrait de ses données...*

- How do we interpret *peut*?
  - $p = \text{démander le retrait...} : \forall \square \exists \Diamond p$.

**Strategy to win a game**

Black has a strategy to put the game in a situation from which White king will never get close to Black pawn.

- $q = \text{White king never gets close to Black pawn} : \exists \Diamond \forall \Box q$.

Other properties related with choices, like noninterference.
ctl properties on transition systems

bullet Hunter/wolf/goat/cabbage puzzle.

▷ Does the initial state satisfy $\forall \Diamond (h = 1 \land w = 1 \land g = 1 \land c = 1)$?
▷ What is the right property that says that the puzzle has a solution?

bullet Deadlock freedom:

▷ Suppose the states of each process are $p_1, p_2, p_3$, resp. $q_1, q_2, q_3$.
▷ Deadlock freedom, i.e. all computations may progress:

$$\forall \Box \bigvee_{1 \leq i \leq 3} (PC_1 = p_i \land \exists \bigcirc PC_1 \neq p_i) \lor \bigvee_{1 \leq i \leq 3} (PC_2 = q_i \land \exists \bigcirc PC_2 \neq q_i)$$
CTL properties on transition systems

- Hunter/wolf/goat/cabbage puzzle.
  - Does the initial state satisfy $\forall \diamond (h = 1 \land w = 1 \land g = 1 \land c = 1)$?
  - What is the right property that says that the puzzle has a solution? $\exists \diamond (h = 1 \land w = 1 \land g = 1 \land c = 1)$

- Deadlock freedom:
  - Suppose the states of each process are $p_1, p_2, p_3$, resp. $q_1, q_2, q_3$.
  - Deadlock freedom, i.e. all computations may progress:

    $$\forall \Box \bigvee_{1 \leq i \leq 3} (PC_1 = p_i \land \exists \bigcirc PC_1 \neq p_i) \lor \bigvee_{1 \leq i \leq 3} (PC_2 = q_i \land \exists \bigcirc PC_2 \neq q_i)$$
CTL properties on transition systems

- Hunter/wolf/goat/cabbage puzzle.
  - Does the initial state satisfy $\forall \Diamond (h = 1 \land w = 1 \land g = 1 \land c = 1)$?
  - What is the right property that says that the puzzle has a solution? $\exists \Diamond (h = 1 \land w = 1 \land g = 1 \land c = 1)$

- Deadlock freedom:
  - Suppose the states of each process are $p_1, p_2, p_3$, resp. $q_1, q_2, q_3$.
  - Deadlock freedom, i.e. all computations may progress:

$$\forall \Box \bigvee_{1 \leq i \leq 3} (PC_1 = p_i \land \exists \Diamond PC_1 \neq p_i) \lor \bigvee_{1 \leq i \leq 3} (PC_2 = q_i \land \exists \Diamond PC_2 \neq q_i)$$
Sample tautologies

- **Tautology**: formula that is true regardless of the truth values given to the atomic propositions.

- **Examples**:

  \[
  \neg \forall \diamond p \leftrightarrow \exists \diamond \neg p \\
  \forall \diamond p \rightarrow \forall \lozenge p \\
  \exists \lozenge \exists \lozenge p \rightarrow \exists \lozenge p \\
  \forall \square (p \land q) \leftrightarrow \forall \square p \land \forall \square q \\
  (\exists \lozenge p \rightarrow \exists \lozenge q) \rightarrow \exists \lozenge (p \rightarrow q)
  \]

- **Formulas which are not tautologies**:

  \[
  \forall \lozenge (p \lor q) \leftrightarrow \forall \lozenge p \lor \forall \lozenge q
  \]

- **To prove they are not tautologies, give a counter-model!**
Minimal set of operators

All CTL formulas can be expressed using the following set of operators:

- Boolean operators (further reducible, e.g., to $\land$ and $\neg$).
- $\forall \Diamond$.
- $\forall U$.
- $\forall \Box$.

Examples – express the following:

- $\exists (p U q)$.
- $\exists \Box p$.

The dual set of path-temporal operators can also be used as minimal set of operators!
Other (linear) temporal operators: weak until, release

- **Weak until** $p \mathcal{W} q$: $p \mathcal{W} q \equiv p \mathcal{U} q \land \Box p$.
- **Release** $p \mathcal{R} q$: $p \mathcal{R} q \equiv \neg (\neg p \mathcal{U} \neg q)$.
- Can be extended to CTL operators: $\forall p \mathcal{W} q$, $\exists p \mathcal{R} q$, etc.
Fixpoints

Globally, forward, until, release can be defined “inductively”:

\[\exists \Diamond p \equiv p \lor \exists \Box \exists \Diamond p\]
\[\forall \Diamond p \equiv \ldots ?\]
\[\exists \Box p \equiv \ldots ?\]
\[\forall \Box p \equiv \ldots ?\]
\[\exists p U q \equiv q \lor (p \land \exists \Box (p U q))\]
\[\forall p U q \equiv \ldots ?\]
\[\exists p R q \equiv q \land (p \lor \Box \exists (p R q))\]
Remarks on LTL vs. CTL (to be continued!)

- Both LTL and CTL formulas are interpreted over transition systems.
- An LTL formula speaks about what happens on one run that starts in a state.
  - Time passage is determined by some superior entity, choices do not exist and no dilemma about possible continuations exists.
  - A posteriori analysis of the behavior of a system (but behaviors may be infinite!).
- A CTL formula speaks about what could happen in various runs that starts in a state.
  - Time is nondeterministic and choices must be taken into account, good/bad things may happen due to good/bad decisions and continuations depend on them.
  - A priori analysis of the possible evolution of a system.
- Some LTL formulas (but not all!) can be represented as CTL formulas:
  - Checking $\Box p$ holds at a state $q$ in a transition system requires checking that all runs starting in $q$ satisfy $\Box p$.
  - Hence, from this state-centered point of view, checking $\Box p$ amounts to checking $\forall \Box p$.
  - No longer holds for more complex formulas!
  - Simply because $\forall (\Diamond p \land \Box q)$ is not a CTL formula!
The model-checking problem

- Given a CTL formula $\phi$ and a finitely presentable model $M$, does $M \models \phi$ hold?
  - Finitely presentable tree = transition system over $AP$.
  - The tree = the unfolding of $A$.

- Note the difference with LTL models:
  - A transition system embodies an uncountable set of models for LTL!
  - A transition system embodies a unique model for CTL!
Which state satisfies $\exists \lozenge p$?

- Search for a reachable state labeled with $p$.

Which state satisfies $\exists \square p$?

- Search for a reachable strongly connected set labeled with $p$.
- Only states in this SCC satisfy $\exists \square p$. 
State labeling algorithm:

- Given formula $\phi$, **split** $Q$ into $Q_\phi$ and $Q_{\neg \phi}$
- Structural induction on the syntactic tree of $\phi$.
- Add a new propositional symbol $p_\phi$ for each analyzed $\phi$.
- Label $Q_\phi$ with $p_\phi$ and do not label $Q_{\neg \phi}$ with $p_\phi$. 
CTL model-checking (2)

For $\phi = \forall \Box p$

- $Q_{\forall \Box p} = \{ q \in Q \mid \forall q' \in \delta(q), p \in \pi(q') \}$
- $Q_{\neg \forall \Box p} = \{ q \in Q \mid \exists q' \in \delta(q), p \notin \pi(q') \}$

Example...
CTL model-checking (3)

\[ \phi = \exists \Box p. \]

- \( Q_{\exists \Box p} \) contains state \( q \) iff \( q \) is labeled with \( p \) and belongs to a circuit containing only \( p \) states.
- \( Q_{\neg \exists \Box p} = Q \setminus Q_{\exists \Box p} \).

Example...
CTL model-checking (4)

\[ \phi = \exists (p_1 \cup p_2) \]

- \( Q_{\exists (p_1 \cup p_2)} \) contains state \( q \) iff \( \exists q' \in Q \) s.t.:

- \( Q_{\neg \exists (p_1 \cup p_2)} = Q \setminus Q_{\exists (p_1 \cup p_2)}. \)

Example...
CTL model-checking example

$$\exists \square p$$
$$\exists p \mathbin{\lor} q$$

$$\forall \square p$$
$$\forall p \mathbin{\lor} q$$

$$\exists \bigcirc \forall \bigcirc p$$
Properties of the (first variant of the) model-checking algorithm

- It seems that the model-checking algorithm requires graph algorithms
  - Successors for $\exists \bigcirc$.
  - Reachability analysis for $\exists U$.
  - Circuits for $\exists \Box$.

- But could we take advantage of the fixpoint expansions of the temporal operators?

\[
\begin{align*}
\exists \Box p & \equiv p \land \exists \bigcirc \exists \Box p \\
\exists p U q & \equiv q \lor (p \land \exists \bigcirc(p U q))
\end{align*}
\]
Fixpoint variant of the model-checking algorithm

- Given a formula $\phi$ and a transition system $M = (Q, q_0, \delta)$,
- ... denote $Sat_M(\phi)$ the set of states in $Q$ which satisfy $\phi$.
- ... and denote $post(q) = \{ r \in Q \mid (q, r) \in \delta \}$.

Theorem

- $Sat(\exists(\phi \cup \psi))$ is the smallest subset $T$ of $Q$ such that:
  1. $Sat(\psi) \subseteq T$ and
  2. If $q \in Sat(\phi)$ and $post(q) \cap T \neq \emptyset$ then $q \in T$.
- $Sat(\forall \square \phi)$ is the largest subset $T$ of $Q$ such that:
  3. $Sat(\psi) \supseteq T$ and
  4. If $q \in T$ then $post(q) \cap T \neq \emptyset$.

The last line can also be read as:

  4. For any $q \in Q$, if $post(q) \cap T = \emptyset$ then $q \notin T$. 
Fixpoint variant of the model-checking algorithm

How to compute $\text{Sat}(\exists (\phi \mathcal{U} \psi))$:
1. Start with $T = \text{Sat}(\psi)$.
2. Append $q$ to $T$ if $q \in \text{Sat}(\phi)$ and $\text{post}(q) \cap T \neq \emptyset$.
3. .... until $T$ no longer grows.

How to compute $\text{Sat}(\exists \Box \phi)$:
1. Start with $T = \text{Sat}(\phi)$.
2. Eliminate, inductively, from $T$ all states for which $\text{post}(q) \cap T = \emptyset$.
3. ... until $T$ no longer diminishes.

Examples....
Fixpoint variant of the model-checking algorithm

\[ \exists (\exists q \cup \forall p) \]

- Compute \( \text{Sat}(\exists q) \).
- Compute \( \text{Sat}(\forall p) \).
- Instantiate \( T = \text{Sat}(\forall p) \).
- Append \( st \) to \( T \) if \( st \in \text{Sat}(\exists q) \) and \( \text{post}(st) \in T \).
Fixpoint variant of the model-checking algorithm

\exists (\exists \bigcirc q \cup \forall \bigcirc p) \quad \forall (\exists \bigdiamond p \cup \exists \Box q)
post and pre

How to compute $Sat(\exists \phi \cup \psi)$:
1. Start with $T = Sat(\psi)$.
2. Append $q$ to $T$ if $q \in Sat(\phi)$ and $post(q) \cap T \neq \emptyset$.
3. The same with $T := pre(T) \cap Sat(\phi)$.
4. Here $pre(T) = \{q \mid \exists r \in Q, (q, r) \in \delta\}$.

How to compute $Sat(\exists \Box \phi)$:
1. Start with $T = Sat(\phi)$.
2. Eliminate, inductively, from $T$ all states for which $post(q) \cap T = \emptyset$.
3. The same with $T := \overline{pre}(T) \cap T$
4. Here $\overline{pre}(T) = Q \setminus pre(Q \setminus T)$.
5. In other words, $\overline{pre}(T)$ contains all the states whose successors all belong to $T$. 
**post and pre**

How to compute $\text{Sat}(\exists \phi \cup \psi)$:

1. Start with $T = \text{Sat}(\psi)$.
2. Append $q$ to $T$ if $q \in \text{Sat}(\phi)$ and $\text{post}(q) \cap T \neq \emptyset$.
3. The same with $T := \text{pre}(T) \cap \text{Sat}(\phi)$.
4. Here $\text{pre}(T) = \{q \mid \exists r \in Q, (q, r) \in \delta\}$.

How to compute $\text{Sat}(\exists \Box \phi)$:

1. Start with $T = \text{Sat}(\phi)$.
2. Eliminate, inductively, from $T$ all states for which $\text{post}(q) \cap T = \emptyset$.
3. The same with $T := \overline{\text{pre}}(T) \cap T$
4. Here $\overline{\text{pre}}(T) = Q \setminus \text{pre}(Q \setminus T)$.
5. In other words, $\overline{\text{pre}}(T)$ contains all the states whose successors all belong to $T$. 